cse 102 11-8-22

- mid 2 : Thur Nov 17

Coin changing Problem:

create a table

$C[i,j]$ = the min # of coins needed to pay j units using only coins in $(d_1, d_2, \ldots, d_i)$

where $1 \le i \le n$ and $0 \le j \le N$.

note:

$c[i, 0] = 0$ for $1 \leq i \leq n$

---

note:

we have 2 options for $c[i, j]$:

(1) use no coins of type i. In this case $c[i, j]$ is $c[i-1, j]$

(2) use at least one coin of type i. In this case $c[i, j]$ is $1 + c[i, j - d_i]$

Thus

$$C[i,j] = \min\left(C[i-1,j],\ 1+C[i,j-d_i]\right)$$

<u>note</u>: if $i=1$ or $j<d_i$ then one value falls outside table.

think of these 'out of bounds' entries as ∞.

if both are out of bounds, then $C[i,j] = \infty$.

see examples.

<u>Exercise</u>:

- write pseudo-code for a recursive algorithm and Print out an optimal set of coins.
- Print out <u>all</u> optimal coin lists
- solve Problem when their are limits on # coins in each type.

__Ex.__ Discrete Knapsack Problem

Thief wishes to steal n objects labeled $1, 2, \ldots, n$. let

$$v_i = \text{value of obj } i$$
$$w_i = \text{weight " " "}$$

The thief will place goods in a knapsack with capacity W

define $(x_1, x_2, \ldots, x_n)$ with

$$x_i = \begin{cases} 1 & \text{obj } i \text{ is stolen} \\ 0 & \text{" " " not stolen} \end{cases}$$

Problem: find $x \in \{0,1\}^n$ s.t.

maximize: $\sum_{i=1}^{n} x_i v_i$

subject to

constraint: $\sum_{i=1}^{n} x_i w_i \leq W$

Two questions

- find <u>value</u> of opt. soln: the max value $\sum x_i v_i$ of goods to stolen
- find an opt. soln: exactly which objects should be stolen, i.e. find $x = (x_1, x_2, \ldots, x_n)$.

any x satisfying

$$\sum_{i=1}^{n} x_i w_i \leq W$$

is called a <u>feasible soln.</u>

define a table $V[1..n; 0...W]$

$V[i,j]$ = max value of objects in set $\{1,..,i\}$ whose total weight is $\leq j$

Two alternatives.

(1) do not include obj i. in this case $V[i,j]$ is $V[i-1,j]$

(2) do include obj i. In this case $V[i,j]$ is

$$v_i + V[i-1, j-w_i]$$

choose

$$V[i,j] = \max\left(V[i-1,j],\ v_i + V[i-1, j-w_i]\right)$$

Include boundary & out of bounds values

$$V[i,j] = \begin{cases} 0 & i=0 \text{ and } j \geq 0 \\ \max\left(V[i-1,j],\ v_i + V[i-1,j-w_i]\right) & i>0 \text{ and } j \geq 0 \\ -\infty & j<0 \text{ or } i<0 \end{cases}$$

Principle of optimality

every optimal instance soln. is a combination of optimal sub-instance solns.

Also called <u>optimal sub-structure</u>.